GeekBrains

**Разработка мобильного приложения мультиформатного сканирования штрихкодов на платформе Android**

Программа: «JAVA Программист»

Специализация: «Цифровые профессии»

Филиппов Михаил Валерьевич

Новосибирск

2024

**Содержание**

Оглавление

[Введение 4](#_Toc159803512)

[Тема проекта 4](#_Toc159803513)

[Цель 4](#_Toc159803514)

[Какую проблему решает 4](#_Toc159803515)

[Задачи 4](#_Toc159803516)

[Инструменты 5](#_Toc159803517)

[Состав команды 5](#_Toc159803518)

[Теоретическая глава 6](#_Toc159803519)

[Что такое штрих код? 6](#_Toc159803520)

[Типы штрих-кодов 7](#_Toc159803521)

[Примеры одномерных кодов 8](#_Toc159803522)

[Примеры двумерных кодов 9](#_Toc159803523)

[История возникновения 11](#_Toc159803524)

[Основные разновидности штрих-кодов 12](#_Toc159803525)

[Области применения 14](#_Toc159803526)

[Сканеры для распознавания штрих-кода 15](#_Toc159803527)

[Необходимость и цель разработки приложения 16](#_Toc159803528)

[Практическая часть 17](#_Toc159803529)

[Обзор и анализ существующих решений 17](#_Toc159803530)

[Требования к приложению 19](#_Toc159803531)

[Проектирование приложения 21](#_Toc159803532)

[1. Архитектура приложения 21](#_Toc159803533)

[2. Выбор инструментов и технологий 21](#_Toc159803534)

[3. Проектирование пользовательского интерфейса 22](#_Toc159803535)

[4. Система навигации по приложению 22](#_Toc159803536)

[Разработка приложения 23](#_Toc159803537)

[1. Настройка среды разработки 23](#_Toc159803538)

[2. Обеспечение интеграции данных 24](#_Toc159803539)

[3. Разработка функциональных модулей 24](#_Toc159803540)

[4. Интеграция модулей в единую систему 24](#_Toc159803541)

[5. Компиляция и сборка 25](#_Toc159803542)

[6. Использование инструментов оптимизация и мониторинга 27](#_Toc159803543)

[Тестирование приложения 30](#_Toc159803544)

[1. Методики тестирования 30](#_Toc159803545)

[2. Описание тестовых сценариев 30](#_Toc159803546)

[3. Результаты тестирования и их анализ 30](#_Toc159803547)

[4. Устранение выявленных недостатков 30](#_Toc159803548)

[Описание возможностей приложения 32](#_Toc159803549)

[1. Работа с разными типами штрихкодов 32](#_Toc159803550)

[2. Интерфейс пользователя и его удобство 32](#_Toc159803551)

[3. Примеры использования приложения в реальных условиях 32](#_Toc159803552)

[Особенности и преимущества 34](#_Toc159803553)

[Стилизация и внешний вид 35](#_Toc159803554)

[Внедрение и эксплуатация приложения 38](#_Toc159803555)

[1. Инструкция по эксплуатации приложения 38](#_Toc159803556)

[2. Рекомендации по внедрению в бизнес-процессы 38](#_Toc159803557)

[3. Оценка влияния приложения на эффективность бизнес-операций 38](#_Toc159803558)

[4. Обеспечение безопасности и сохранности данных в возможных сценариях применения 39](#_Toc159803559)

[Обзор полученных результатов 40](#_Toc159803560)

[1. Анализ соответствия разработанного приложения первоначальным требованиям 40](#_Toc159803561)

[2. Финансовые выгоды от замены профессионального сканерного оборудования на приложение 40](#_Toc159803562)

[3. Оценка удовлетворенности потребителей 40](#_Toc159803563)

[4. Гибкость настройки пользовательского интерфейса 41](#_Toc159803564)

[Выводы и предложения по дальнейшему развитию 42](#_Toc159803565)

[Заключение 43](#_Toc159803566)

[Список используемой литературы 45](#_Toc159803567)

[Приложения 46](#_Toc159803568)

[Характеристики персонального компьютера, на котором выполнялась разработка 46](#_Toc159803569)

[Характеристики смартфона, используемого для разработки и тестирования 48](#_Toc159803570)

[Ключевые возможности библиотеки Google ML Kit 50](#_Toc159803571)

# Введение

## Тема проекта

Разработка мобильного приложения мультиформатного сканирования штрихкодов на платформе Android.

## Цель

Основная цель данного дипломного проекта заключается в создании мультифункционального мобильного приложения, предназначенного для распознавания и обработки разнообразных форматов штрихкодов с использованием возможностей платформы Android. Это приложение направлено на демонстрацию возможности обеспечения пользователей надёжным и удобным инструментом для быстрого и точного считывания информации с штрихкодов, способного адаптироваться к широкому спектру потребностей как в личном использовании, так и в бизнес-задачах.

## Какую проблему решает

Проект реализуется с целью решения ряда проблем, среди которых недостаток универсальности существующих сканеров штрихкодов, ограниченная функциональность приложений сканирования доступных на рынке, высокий порог вхождения для пользователей и отсутствие возможности работы с несколькими форматами штрихкодов в рамках одного приложения. Также приложение ставит своей задачей устранение трудностей, связанных с интеграцией сканера штрихкодов в различные бизнес-процессы и повседневную жизнь пользователей.

## Задачи

1. Исследование существующих решений в области мобильных приложений для сканирования штрихкодов.
2. Анализ требований к функционалу мультиформатного сканирования штрихкодов для мобильных устройств.
3. Разработка функциональных требований и спецификаций приложения.
4. Проектирование и создание пользовательского интерфейса, ориентированного на удобство и интуитивную понятность в использовании.
5. Изучение и применение на практике библиотеки Google ML Kit.
6. Интеграция в приложение алгоритмов распознавания и обработки различных видов штрихкодов.
7. Реализация основного функционала приложения.
8. Тестирование приложения и его адаптация для корректной работы на различных устройствах.
9. Разработка документации и инструкций для конечных пользователей приложения.
10. Оценка эффективности приложения и его вклад в решение поставленной проблемы.

## Инструменты

* Android Studio 2023.1.1 Patch 2
* Google ML Kit
* Gradle
* Git
* Android 14

## Состав команды

Филиппов Михаил Валерьевич (разработчик)

# Теоретическая глава

## Что такое штрих код?

Штрих-код – графическая информация, наносимая на поверхность, маркировку или упаковку изделий, предоставляющая возможность считывания её техническими средствами. В нанесении используются белые и черные цвета. Такое изображение наносят на товар перед продажей.

Путем сканирования штрих-кода можно получить всю надлежащую информацию о продукции. У каждого кода имеется свое уникальное изображение, в котором зашифровано все то, что захотел донести производитель до конечного потребителя. Существует несколько видов штриховых кодов:

По типу изображения разделяют на две категории:

* Одномерные (1D), у таких имеются только вертикальные полосы;
* Двухмерные (2D), в таких идет сочетание вертикальных полос с другими геометрическими фигурами, например, точками.

## Типы штрих-кодов

Типы одномерных кодов

* UPC/EAN-128;
* EAN-13;
* UPC-E;
* Code39;
* UPC-A;
* EAN-8;
* «Interleaved 2of5».

### Примеры одномерных кодов

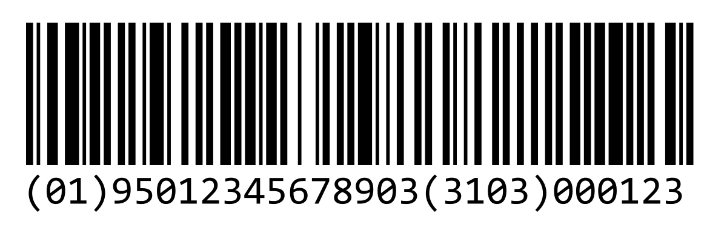


Рис 1. EAN-128

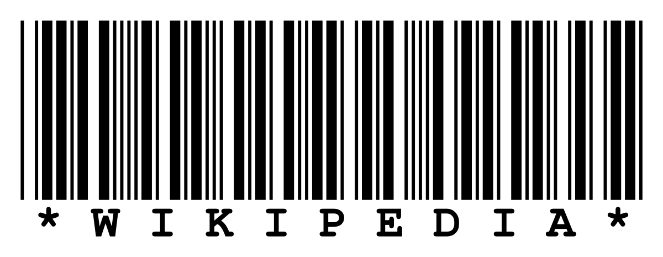


Рис 2. Code39

Безусловно от стандарта производства будут зависеть и типы штрих-кодов. Большую популярность имеют тринадцати символьные EAN-коды. Они используются в пищевой промышленности.

Типы двухмерных кодов

* QR код;
* PDF417;
* Microsoft Tag;
* MaxiCode;
* Aztec Code;
* DataMatrix.

### Примеры двумерных кодов

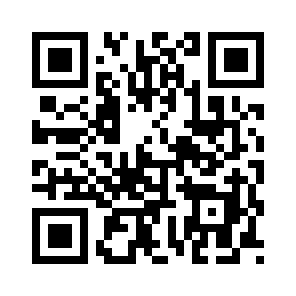


Рис 3. QR-код

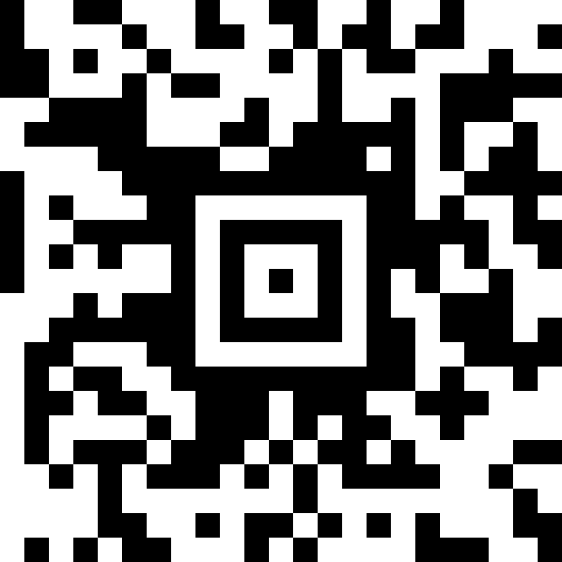


Рис 4. Код AZTEC

QR-коды используются для передачи данных между устройствами, их область применения гораздо шире. QR-коды имеют свою значимость не только в промышленности или в производстве. Также они широко используются в социальной жизни людей, например, в социальных сетях.

Использование штрихового кодирования широко распространено на производстве с большим количеством разного вида выпускаемой продукции. Такая технология уменьшает риск человеческой ошибки ручного набора в несколько раз. Помогает быстро обработать информацию на складе, в логистике и непосредственно в торговле. Для создания кода существуют специальные компьютерные программы и принтеры этикеток, а для их считывания – сканеры штрих-кода.

## История возникновения

Американское производство и промышленность в 40-х годах прошлого века перестала справляться с информацией, поступающих от маркетологов. Эту проблему взяли на себя исследователи Норман Джозеф Вудланд и Бернард Сильвер. Достаточно длительная совместная работа привела к отличному результату. Они разработали штриховую маркировку. Первый штрих-код состоял из точек и тире азбуки Морзе. Чтобы считать информацию, они заимствовали технологию из киноиндустрии – оптическую звукозапись. Такой способ обработки данных прошел все возможные проверки и был запатентован в 1952 году. Так начался путь совершенствования.

В настоящее время принцип штрихового кодирования остался тот же, но вот визуально он сильно изменился.

## Основные разновидности штрих-кодов

Штрих-коды подразделяются на линейные и двухмерные. Их различие в изображении и вместительности данных. Выбор будет зависеть от той информации, которая там будет храниться.

Линейные коды обозначают – 1D, они представляют собой черные вертикальные полосы с разными пробелами на белом фоне. Полосы все разной ширины. Вместимость у одномерных кодов малая, не больше 30 символов. Чаще всего штрих-код несет в себе информацию об артикуле товара или серийном номере и считывается по горизонтали. Считать такой код очень просто. С этой задачей справляются одномерные 1D сканеры.

Для удобства разработали международные стандарты кодирования – EAN.

Состоять такой код может или из 8, или из 13 символов:

* первые 2-3 цифры содержат название страны происхождения товара;
* после них идут 4-5 цифр, они несут информацию, касающуюся изготовителя продукции;
* следующие порядковые 5 цифр – информация о самом товаре;
* крайние 1-2 цифры контролируют правильность всех данных в целом.

Двухмерные коды (2D) не получится считать по горизонтали, их считывают и по горизонтали, и по вертикали. В их структуру входит разнонаправленные линии, также присутствуют точки, клетки и даже другие фигуры.

Двухмерный штрих-код состоит из матричных и многоуровневых кодов. Происходит «склейка» информации, запечатанной в фигурах, что дает сохранить гораздо больше информации. Такой код может уместить в себе около 4 тысяч символов. В двухмерный штрих-код можно сохранить полную информацию о товаре или продукции, включая все характеристики, состав, указать страну происхождения.

Популярность получили 3 вида кодов – Data Matrix, QR-код, и Aztec.

* Data Matrix. Код имеет переменную длину, выполнен в виде матрицы. Этот тип состоит из точек белого и черного цвета разного размера. В Data Matrix встроена система, позволяющая корректировать ошибки и восстанавливать 30% данных с поврежденных участков кода.
* QR-код. Такой код гордость японской компании Denso Wave. Узнать QR-код очень легко, у него имеется 3 квадрата по углам. Передает информацию мгновенно. При повреждении кода имеется возможность восстановить хранившуюся информацию до 30%.
* Код Aztec. Внешне имеет сходство с мишенью. В центре находится квадрат. Слои данных зашифрованы в белых и черных точках. Особым преимуществом является практически 100% восстановление информации с поврежденного кода.

## Области применения

Области применения обширны из-за универсальности штрих-кодов. Их активно используют и в легкой, и в тяжелой промышленности. Но определенные стандарты шифрования уже по традиции связывают с конкретными областями производства.

* EAN широко распространен в торговле, чаще всего встречается на весовом товаре. Например, фруктах, конфетах.
* Разные виды EAN-кодов предназначены для определенной среды. ISBN отлично подходит для маркировки книг, а вот ISSN чаще всего используют в печатной продукции, которая выходит на прилавки регулярно. Фармакод, соответственно, разработан специально под фармацевтические компании.
* Data Matrix прочно осел на рынке компьютерно-вычислительной техники. Но также этими кодами пользуются производители автомобилей.
* QR-код хорошо показал себя в маркетинге, он сейчас на пике популярности. Его также применяют для маркировки табачной и алкогольной продукции.
* Aztec занял нишу в грузоперевозках, популярен в ж/д перевозках. Также этот вид штрих-кодов используют на почте и в других сферах, касающихся транспортировки различных грузов.

## Сканеры для распознавания штрих-кода

При сканировании кода, происходит перевод изображения в последовательность цифр и других значений, а те в свою очередь, уже отображаются в компьютере или в другом устройстве той информацией, что заложили изначально в штрих-код.

В сканере бывают несколько видов считывающих элементов – светодиодные, имидж-сканеры и лазерные. В светодиодном сканере считывание информации происходит благодаря ПЗС-матрице, а подсвечивание происходит за счет светодиода. Сканеры с такой системой достаточно бюджетные, но качественные. Подходят для работы с линейными штрих-кодами.

В лазерных сканерах имеется узкий лазерный луч, с помощью которого считывать информацию можно на разном расстоянии. Для обычных моделей расстояние может достигать даже 30 сантиметров. А для промышленных до 19 метров. Такие сканеры не способны прочитать данные с поврежденных этикеток. Их, так же, как и светодиодные, используют для распознавания линейных штрих-кодов.

Имидж-сканеры, а проще фото-сканеры распространены больше всего на данный момент. Они фотографируют штрих-код и отправляют его в базу данных. Им абсолютно без разницы в какому типу относится штрих-код, работают как с двухмерными 2D, так и с одномерными 1D. Очень хорошо распознают даже размытые изображения, не говоря уже о мелких изъянах этикетки. Могут считывать информацию с пластиковых карт и с экранов мобильных телефонов. Они используются не только в сфере производства, но и в маркетинге.

## Необходимость и цель разработки приложения

Одной из ключевых задач является предоставление доступной альтернативы дорогостоящему промышленному оборудованию, используемому для сложных задач сканирования в бизнес-целях. Профессиональные сканеры штрихкодов, хотя и являются надежными и мощными инструментами, часто сопровождаются высокими затратами на приобретение и обслуживание, что может оказаться серьёзной статьей расходов, особенно для малого и среднего бизнеса. В свете этого обстоятельства, целью настоящего проекта ставится не только создание приложения, способного эффективно распознавать широкий спектр штрихкодов, но и демонстрация того, что использование стандартного смартфона, оснащенного данным приложением, может успешно заменить дорогие сканеры в повседневных операциях многих предприятий.

Эта цель особенно актуальна с учетом всё возрастающего распространения мобильных технологий и общей унификации устройств. Недорогие, но мощные смартфоны могут служить эффективным средством для автоматизации бизнес-процессов, повышая таким образом их доступность для широкого круга предприятий. Кроме того, использование мобильного приложения для сканирования штрихкодов значительно снижает порог входа для малых предприятий, стремящихся максимизировать свою операционную эффективность без необходимости инвестировать в специализированное, часто недоступное оборудование. Таким образом, механизмы и реализации, используемые в данном мобильном приложении, могут стать мощным инструментом в малом и среднем бизнесе, помочь оптимизировать рабочие процессы и повысить конкурентоспособность на рынке.

# Практическая часть

## Обзор и анализ существующих решений

В современном мире штрихкоды являются универсальным средством для автоматизации и ускорения процесса идентификации и учета товаров и услуг. С этой целью был разработан широкий спектр сканеров штрихкодов, варьирующихся от ручных устройств до стационарных систем на кассах и специализированных промышленных сканеров. Их функции могут включать проведение инвентаризации, автоматизацию ввода данных, трекинг активов и более того. Тем не менее, многие из этих решений обладают рядом ограничений и недостатков.

Одним из основных недостатков традиционных сканеров является их стоимость, которая может быть значительной, особенно для высокоспециализированного оборудования. Кроме того, многие коммерчески доступные приложения зависят от интернет-соединения для сопоставления штрихкодов с базами данных, что может быть ограничивающим фактором в зонах с плохим покрытием. Более того, их эффективность может значительно снижаться в условиях плохой освещенности, а также они могут быть несовместимы с определёнными редкими или устаревшими типами штрихкодов.

На фоне данных ограничений разрабатываемое приложение предлагает инновационный подход к сканированию штрихкодов на базе машинного обучения. Моё приложение демонстрирует применение технологий искусственного интеллекта для распознавания изображений, что позволяет ему эффективно работать даже в условиях недостаточного освещения и при отсутствии интернет-соединения.

Используя предварительно обученную модель, которая встроена прямо в приложение, реализуется быстрое и точное сканирование широкого спектра типов штрихкодов, что делает его удобным инструментом для любого вида бизнеса без дополнительных капитальных вложений в специализированное оборудование. Модель машинного обучения оптимизирована так, чтобы быть легкой и не требовательной к ресурсам смартфона, что позволяет приложению работать на большинстве устройств без потери производительности.

Данное приложение не просто добавляет удобство в сканирование штрихкодов, но и открывает новые возможности для экономии и оптимизации бизнес-процессов, предоставляя на выходе технологичное решение, доступное каждому пользователю.

## Требования к приложению

1. Функциональные требования:

Сканирование и распознавание штрихкодов: Приложение должно быть способно быстро и точно сканировать и распознавать различные типы штрихкодов, включая, но не ограничиваясь, EAN, UPC, QR-кодами и Code 128.

Работа без интернета: Приложение должно функционировать в автономном режиме, не требуя подключения к интернету для процесса сканирования и распознавания штрихкодов.

Использование предварительно обученной модели ИИ: В приложение должна быть интегрирована обученная модель машинного обучения, что позволит осуществлять распознавание штрихкодов без задержек.

Работа в условиях плохой освещенности: Приложение должно предоставлять высокоточное сканирование даже при слабом освещении.

Поддержка множества устройств: Приложение должно быть совместимо с большинством смартфонов, вне зависимости от их технических характеристик.

Пользовательский интерфейс: Приложение должно иметь интуитивно понятный интерфейс, который позволяет пользователю легко осуществлять сканирование и просмотр информации о распознанных штрихкодах.

2. Нефункциональные требования:

Производительность: Приложение должно обрабатывать сканирование и распознавание штрихкодов быстро, без существенных задержек.

Масштабируемость: Приложение должно быть масштабируемым в плане добавления поддержки новых типов штрихкодов и интеграции с другими системами и устройствами.

Устойчивость и надежность: Приложение должно стабильно функционировать в течение длительного времени, минимизируя случаи сбоев или ошибок.

Безопасность: Приложение должно защищать любые персональные данные пользователей и не допускать их несанкционированное использование.

Удобство использования: Приложение должно быть простым в использовании для конечных пользователей без необходимости прохождения сложного обучения.

Совместимость: Приложение должно быть корректно запускаться на основных операционных системах смартфонов.

Поддержка и обновления: Обеспечение регулярной поддержки и обновлений приложения для улучшения его функциональности и безопасности.

Эти требования не только определяют технические аспекты вашего приложения, но и устанавливают ожидания для конечного продукта, чтобы удовлетворить потребности пользователей и бизнес-целей.

## Проектирование приложения

### 1. Архитектура приложения

Приложение будет следовать одностраничной архитектуре с минималистичным дизайном. Архитектурный подход фокусируется на удобстве использования и быстроте работы. Основой служит разделение на два основных компонента: пользовательский интерфейс (UI) и модуль распознавания штрихкодов. UI отвечает за взаимодействие с пользователем, а модуль распознавания - за обработку данных с камеры и интерпретацию штрихкодов. Вся логика работы с ИИ инкапсулирована внутри модуля распознавания для упрощения модификации и масштабирования.

На одноэкранном интерфейсе отобразятся не только данные, полученные со сканированного штрихкода, но и дополнительные детали, такие как формат штрихкода и тип закодированной информации (например, веб-ссылка, простой текст, электронная почта, телефонный номер и т.п.).

Функциональность распознавания штрихкодов в мобильном приложении будет включать следующее:

* Определение формата штрихкода: после сканирования штрихкода, приложение автоматически определяет его формат (например, UPC, EAN, QR-код и т.д.) и отображает эту информацию в текстовом поле на экране.
* Распознавание типа содержимого: Приложение анализирует данные внутри штрихкода, чтобы определить тип содержимого: это может быть URL веб-сайта, простой текстовый фрагмент, адрес электронной почты, номер телефона и т.д. Эта информация затем отображается пользователю для удобства.

### 2. Выбор инструментов и технологий

Для разработки приложения выбраны инструменты, которые обеспечивают простоту и быстродействие. Используется кроссплатформенный фреймворк, позволяющий запускать приложение на основных ОС смартфонов без значительного переписывания кода. Модуль ИИ основан на облегченной нейронной сети, способной работать на устройствах с ограниченными ресурсами, в качестве инструмента машинного обучения используется TensorFlow Lite.

### 3. Проектирование пользовательского интерфейса

Интерфейс приложения представлен одним экраном с минимальным набором элементов. В центре экрана размещается кнопка "Сканировать", а над ней - текстовое поле, в которое выводится информация о распознанном штрихкоде после сканирования. Пользовательский интерфейс спроектирован так, чтобы быть интуитивно понятным для пользователя, минимизировав необходимость ввода и избегая информационного перегруза.

### 4. Система навигации по приложению

Навигация в приложении отсутствует в традиционном понимании, т.к. приложение состоит из одного экрана. Взаимодействие пользователя с приложением сводится к нажатию на кнопку "Сканировать", после чего активируется камера устройства для сканирования штрихкода. После успешного сканирования и распознавания штрихкода, пользователь автоматически возвращается на основной экран приложения, где отображается полученная информация.

Такой подход к проектированию делает приложение чрезвычайно легким в использовании, поскольку исключает необходимость навигации по нескольким экранам и вкладкам, сосредотачиваясь исключительно на основной задаче - сканировании и отображении информации о штрихкоде.

## Разработка приложения

### 1. Настройка среды разработки

Для эффективной разработки нашего приложения прежде всего необходимо настроить среду разработки. Была выбрана проверенную и привычную многим Android Studio, поскольку разработка будет вестись под платформу Android. Осуществляем установку всех требуемых плагинов и библиотек, обеспечиваем поддержку версионного контроля через системы, такие как Git, настраиваем эмуляторы для тестирования приложения и обеспечиваем актуальность SDK для выбранной платформы.



Рис 5. Android Studio 2023.1.1 Patch 2

В контексте разработки этого проекта, я ориентируюсь на прямой и ограниченный подход, где весь процесс проектирования, разработки и тестирования осуществляется на одном персональном компьютере. Это исключает необходимость в сложных системах непрерывной интеграции и развертывания (CI/CD) вроде Jenkins, GitLab CI или GitHub Actions. Будут использоваться локальные инструменты и фреймворки, предоставляемые средой Android Studio, включая встроенные средства для отладки и тестирования приложения. Встроенного функционала данной среды разработки вполне достаточно для успешной разработки малых и средних проектов. Такая схема позволяет упростить рабочий процесс, сократить временные затраты на настройку внешних систем и полностью контролировать цикл разработки без дополнительных звеньев. Это поможет нам оставаться максимально гибкими и быстро адаптироваться под изменяющиеся требования проекта.

### 2. Обеспечение интеграции данных

Поскольку приложение не требует сложного хранения данных и оперирует заранее определенными структурами (форматы штрихкодов и типы данных), создание традиционной базы данных не применимо. Вместо этого в код встраивается необходимый минимальный набор правил и определений, которые обеспечивают распознавание и классификацию данных, получаемых посредством использования ML Kit.

### 3. Разработка функциональных модулей

Функциональный модуль распознавания штрихкодов является центральным и единственным в структуре приложения. В процессе разработки этот модуль рассматривается как самостоятельная компонента, которая включает комплексную логику работы с ML Kit. В модуле внедряется интеграция с камерой устройства для захвата изображения и последующей обработки с помощью API ML Kit. Рассмотрение различных сценариев использования и соответствующих им обработчиков данных гарантирует гибкость и расширяемость приложения.



Рис 6. Библиотека Google ML Kit

### 4. Интеграция модулей в единую систему

Интеграция единственного функционального модуля с пользовательским интерфейсом проходит в рамках единой функциональной среды, обеспечивая плавное и нативное взаимодействие. Благодаря такому подходу реализуется простой и интуитивно понятный пользовательский поток с минимальными задержками в работе. Под логическую интеграцию подпадает не только связь между кнопкой "Сканировать" и модулем камеры, но и последующая обработка сканированных данных с выводом информации на экран устройства.

Такой подход к разработке, где отсутствует излишняя сложность и множество взаимосвязанных компонентов, говорит о грамотном планировании и ориентации на удобство конечного пользователя. Приложение, сконцентрировавшее свою функциональность вокруг одного модуля, демонстрирует принципиальную преданность простоте и надежности.

### 5. Компиляция и сборка

Для эффективного управления зависимостями в проекте был использован инструмент Gradle, что позволило автоматизировать процессы сборки проекта и управлять библиотеками. Контроль версий и совместимость компонентов системы тщательно отслеживались для предотвращения конфликтов и обеспечения стабильности зависимостей.

Применение Gradle значительно упростило многие аспекты разработки. Как мощная система автоматизации, основанная на принципах конфигурации как кода, Gradle предоставил следующие преимущества:

* Модульность и Многозадачность. Gradle позволил развивать проект в модульной манере. Благодаря его способности обрабатывать множество проектов в рамках одного сборочного процесса, мы добились более высокой организации кода и эффективности в выполнении задач. Это особо ценно при работе в больших командах, где задачи могут быть распределены по отдельным модулям.
* Гибкость и Кастомизация. С помощью скриптов сборки на языке Groovy или Kotlin, можно настроить жизненный цикл проекта под конкретные нужды разработки приложения. Это означает, что любой процесс от компиляции исходного кода до его тестирования и сборки можно адаптировать для удовлетворения самых специфических требований.
* Управление Зависимостями. Централизованный механизм управления зависимостями обеспечил однородное и последовательное взаимодействие всех библиотек и инструментов, используемых в проекте. Возможность определения версий зависимостей в одном месте и их автоматическое разрешение экономит время и предотвращает ошибки, связанные с несоответствиями версий.
* Интеграция с IDE и Системами Контроля Версий. Gradle идеально интегрируется с популярными средами разработки, такими как Android Studio, что упрощает процесс разработки и отладки. Поддержка различных систем контроля версий позволяет легко отслеживать изменения и вносить соответствующие обновления в сборочные скрипты.
* Поддержка Непрерывной Интеграции. Gradle совместим с платформами, предоставляющими услуги континуальной интеграции и доставки (CI/CD), такими как Jenkins, Travis CI, и GitLab CI. Это позволяет автоматизировать процесс развертывания и гарантировать, что новый код будет тестироваться и собираться корректно после каждого коммита. Данная возможность необходима для дальнейшего развития проекта и использования этого подхода в коммерческой разработке
* Производительность и Кэширование. Одним из главных преимуществ Gradle является сокращение времени сборки за счет инкрементной компиляции и кэширования. Применение этих механизмов значительно сокращает время ожидания разработчиков, особенно при работе над крупными проектами с многочисленными зависимостями.

На протяжении всего процесса создания приложения, Gradle обеспечил скорость, автоматизацию и контроль, которые важны для современной мобильной разработки. Регулярные обновления и поддержка сообщества делают этот инструмент одним из предпочтительных решений в индустрии и ключевым элементом здорового жизненного цикла проекта разработки программного обеспечения.

### 6. Использование инструментов оптимизация и мониторинга

С помощью Profiler есть возможность наблюдать, как приложение взаимодействует с системными ресурсами в различных состояниях: активном, приостановленном, восстанавливающемся. Это позволило оптимизировать управление ресурсами и сделать работу приложения более плавной и предсказуемой.

Анализ Процессорного Времени даёт возможность анализировать нагрузку на процессор и время, затрачиваемое на выполнение задач по каждому потоку. Используя эту информацию, были оптимизированы потоки выполнения для более равномерного распределения нагрузки.

RAM Profiler даёт нам полную картину использования памяти приложением, что стало ключом к оптимизации и управлению памятью. Были выявлены возможные утечки памяти и избыточные аллокации, повысив тем самым общую производительность и устойчивость.

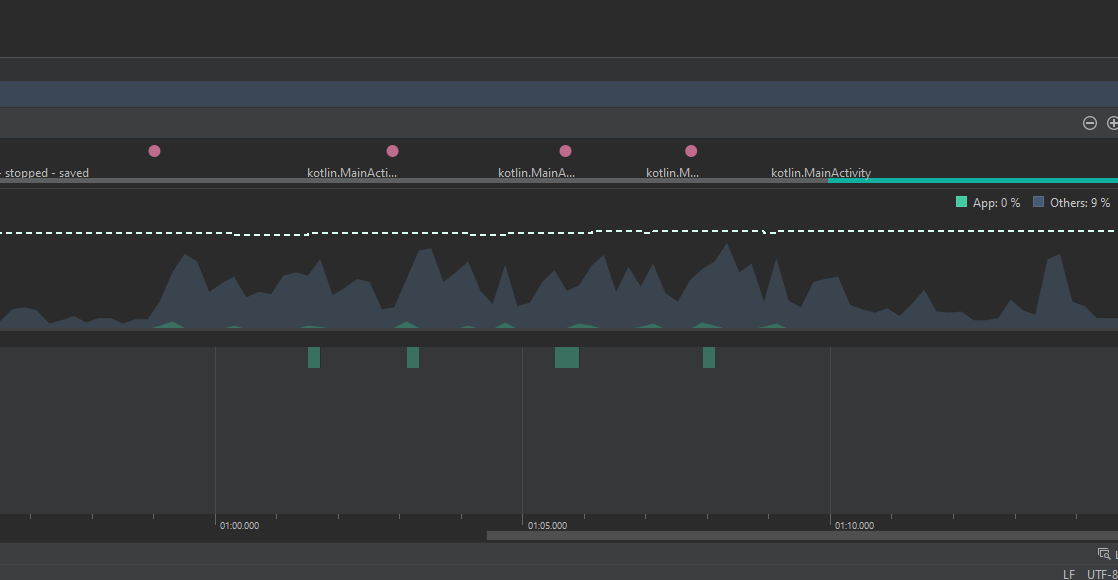


Рис 7. Приложение под нагрузкой

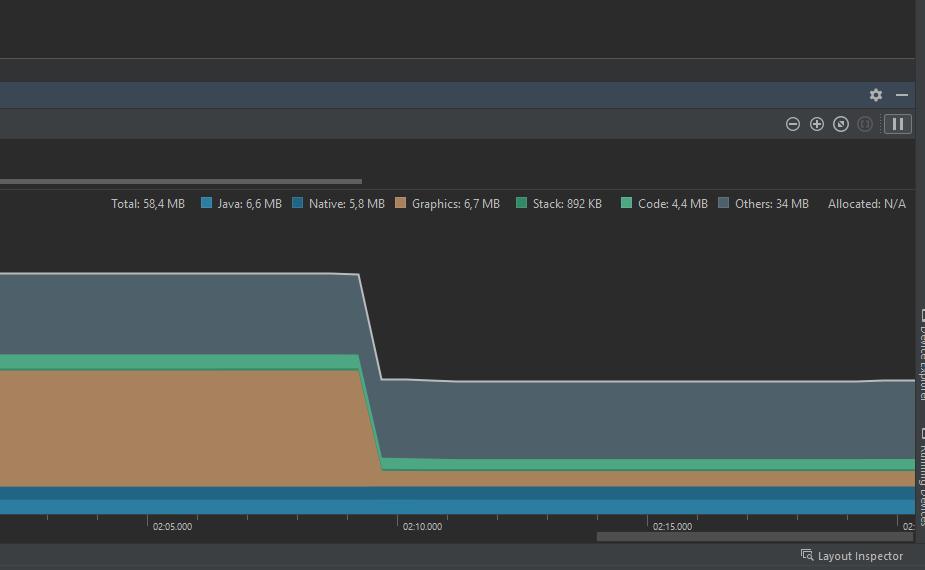


Рисунок 8. Мониторинг состояние памяти

Чтобы оптимизировать приложение с точки зрения энергопотребления, я использовал следующий подход:

* С помощью функции Profileable power rails появилась возможность определить, какие компоненты устройства - такие как CPU, GPU, и экран - влияют на общий расход энергии во время работы приложения. Это позволило мне более целенаправленно оптимизировать использование ресурсов.
* С использованием System Trace было произведено наблюдение записей о потреблении энергии в контексте операций, выполняемых в приложении. Это помогло мне увидеть, как определенные действия или этапы жизненного цикла приложения влияют на энергопотребление и на основе этих данных принимать решения о дальнейших оптимизациях.
* Power Profiler предоставляет графическое представление потребления энергии, что значительно упростило процесс определения проблемных областей. Наглядный анализ показал, что определенные функции сканирования штрихкодов были особенно энергозатратными и требовали доработки.
* Получившиеся данные о расходе энергии позволили провести серию тестирований с целью снижения энергопотребления в различных сценариях использования приложения. Была повышена энергоэффективность работы в фоновом режиме.

Благодаря новому Power Profiler, работа стала более направленной и продуктивной в плане оптимизации энергоэффективности. Мне удалось не просто уменьшить общий расход энергии приложением, но и улучшить его работу на устройствах с ограниченными ресурсами аккумулятора, что в конечном итоге позволит предложить пользователям действительно удобное и экономичное решение.

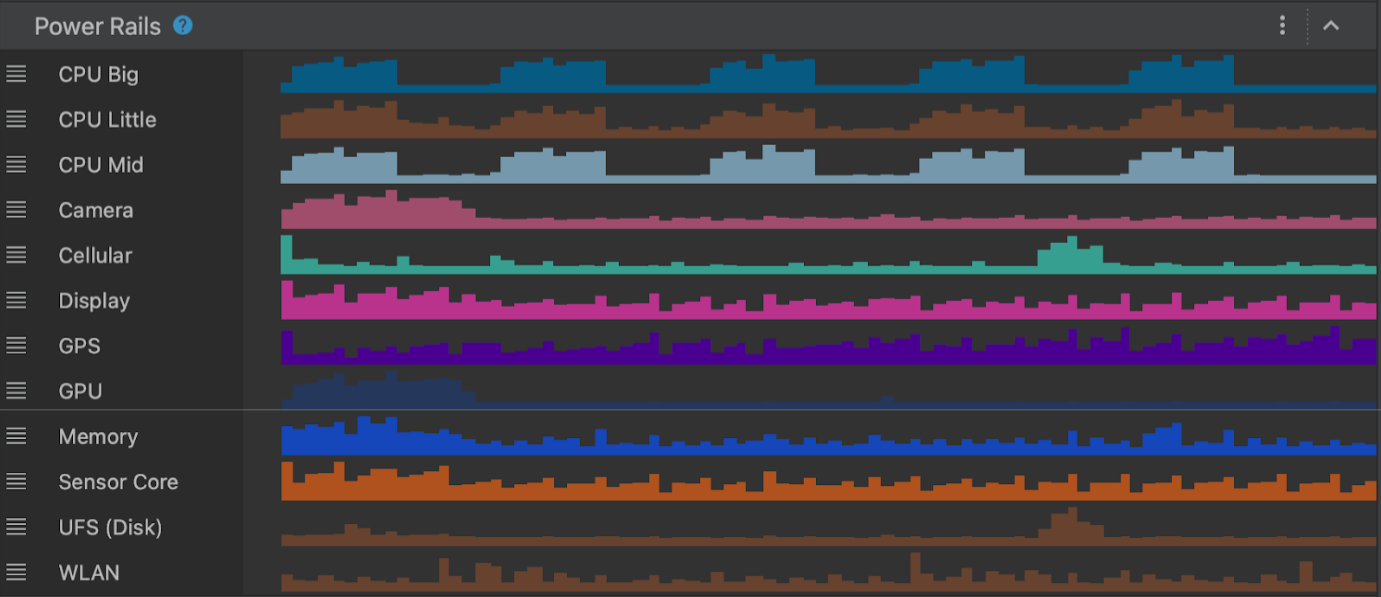


Рис 9. Power Profiler rails

## Тестирование приложения

### 1. Методики тестирования

В ходе разработки приложения я придерживался ручного тестирования, которое оказалось наиболее подходящим для нашего масштаба работы. Преимущество этого подхода заключается в его простоте, легкости реализации и гибкости. Приложение запускалось в различных условиях, чтобы удостовериться в стабильности его функционирования и корректности выполнения основного функционала, такого как распознавание штрихкодов с использованием ML Kit. Это позволило оперативно обнаруживать и исправлять проблемы, адаптировать интерфейс под различные размеры экранов и проверить общую производительность.

### 2. Описание тестовых сценариев

Для тестирования был подготовлен ряд базовых сценариев, которые заключались в проверке функции сканирования штрихкода на различных устройствах и при разной освещённости. Также внимание было уделено проверке, как приложение справляется с распознаванием разнообразных форматов штрихкодов. Особое внимание уделялось юзабилити – приложение должно было быть простым и удобным в использовании.

### 3. Результаты тестирования и их анализ

В ходе тестирования были выявлены некоторые небольшие проблемы с интерфейсом, которые без труда устранялись по мере их обнаружения. Также, благодаря обратной связи от независимых тестировщиков, удалось оптимизировать процесс сканирования для работы на разных устройствах. Различия в производительности устройств и размерах экранов дали важное понимание о том, как моё приложение ведёт себя в реальных условиях использования.

### 4. Устранение выявленных недостатков

Любые недостатки, обнаруженные во время процесса тестирования, были быстро устранены. Это позволило улучшить качество приложения и обеспечить более удобное и стабильное его функционирование. Изначальная стратегия тестирования, хоть и относительно простая, показала свою эффективность, демонстрируя, что даже несложные методы могут быть мощным инструментом в руках разработчика.

## Описание возможностей приложения

Разработанное приложение представляет из себя надёжное средство чтения штрихкодов, которое поддерживает работу с множеством различных типов штрихкодов, обеспечивая универсальность и широкий спектр применения. Это не просто инструмент для сканирования; приложение имеет потенциал к интегрированию в различные бизнес-процессы, от логистики до розничных операций.

### 1. Работа с разными типами штрихкодов

Приложение спроектировано таким образом, чтобы легко и безошибочно считывать штрихкоды различных форматов, включая UPC, EAN, QR-коды и многие другие. Вне зависимости от формата и размеров, приложение быстро и точно проводит сканирование.

### 2. Интерфейс пользователя и его удобство

Ключевой особенностью приложения является интуитивно понятный пользовательский интерфейс, созданный с учётом удобства и эффективности. Внимание было сосредоточено на минимализме и функциональности, чтобы пользователи могли без затруднений использовать приложение без долгой адаптации. Приложению не требуются дополнительные разрешения – это безусловно является преимуществом, поскольку аспекты обеспечения безопасности гарантируются встроенными средствами сервисов операционной системы.

### 3. Примеры использования приложения в реальных условиях

Хотя на первый взгляд приложение может показаться простым средством чтения штрихкодов, его можно эффективно внедрить в бизнес-процессы, например, для учёта товаров на складе или мгновенного доступа к информации о продукте в торговом зале. Оно может служить в роли мостика между физическим товаром и его цифровым представлением в базе данных. Кроме того, простота использования делает его ценным инструментом для персонала, не требуя от них специализированного обучения.

В домашних условиях приложение может быть использовано для управления личными запасами или организации коллекций. Это приложение - продукт, демонстрирующий возможность простого внедрения сканирования штрихкодов в повседневную жизнь и бизнес-операции, предоставляя прочную платформу для дальнейшей настройки и интеграции. Тем не менее, приложение уже само по себе является законченным продуктом, который может быть использован в повседневной жизни.

## Особенности и преимущества

Безопасность и приватность

Разрешения, особенно связанные с доступом к камере устройства, являются чувствительным моментом для конечных пользователей из-за проблем конфиденциальности и безопасности. Современные API, такие как Google Code Scanner API, предлагают элегантные решения этой проблемы. Когда API может функционировать без необходимости запрашивать разрешение на доступ к камере или другим потенциально чувствительным функциям устройства, это уменьшает риск для конфиденциальности и безопасности пользовательских данных.

Преимущество использования таких API заключается в том, что они обеспечивают более высокий уровень доверия со стороны пользователей, поскольку им не приходится предоставлять приложению глубокие уровни доступа к своим личным данным. Это также снижает риск непреднамеренного доступа к данным из-за уязвимостей приложения или устройства - если приложение вообще не запрашивает разрешения, оно не может злоупотреблять ими.

В такой ситуации защищенное API обрабатывает все данные на своей стороне, и приложение получает уже обработанный результат, что обеспечивает дополнительный слой безопасности. Для разработчиков это также упрощает процесс создания и обслуживания приложений, так как им не приходится внедрять и поддерживать механизмы работы с разрешениями.

Безусловно, для некоторых приложений и целей возможность иметь собственный интерфейс и обработку данных непосредственно на устройстве пользователей является необходимой или желательной. В таких случаях разработчикам придется тщательно работать с разрешениями, чтобы обеспечить как безопасность, так и удобство использования.

## Стилизация и внешний вид



Рис 10. Главный экран приложения

В своем дипломном проекте я не просто создал функциональное приложение, но также продемонстрировал важный аспект гибкости в сфере дизайна и брендинга. Стилизация приложения под брендинг GeekBrains служит яркой иллюстрацией того, как продукт может быть адаптирован под визуальную идентификацию любой компании, сохраняя при этом свою основную функциональность и полезность.

Это представляет ценность для потенциальных клиентов и партнеров, поскольку показывает, что дизайн приложения может быть легко кастомизирован для соответствия фирменному стилю клиента, что делает его идеальным решением для интеграции в брендированные стратегии различных организаций.

Персонализация приложения под конкретный бренд не только улучшает пользовательский опыт, делая его более интуитивно понятным и привлекательным для целевой аудитории, но и укрепляет визуальную связь с брендом, что способствует лояльности клиентов и узнаваемости.



Рис 11. Интерфейс режима поиска штрихкодов



Рис 12. Пример успешного нахождения и распознавания штрихкода

## Внедрение и эксплуатация приложения

Для внедрения моего приложения в работу организации потребуется лишь установить его на устройства сотрудников, использующихся для работы со штрихкодами. Эксплуатация приложения не требует специальных знаний или обучения благодаря интуитивно понятному интерфейсу и простоте использования.

### 1. Инструкция по эксплуатации приложения

После установки приложения на устройство пользователь может немедленно начать работу, просто запустив приложение и направив камеру на необходимый штрихкод. Информация автоматически считывается и отображается на экране устройства. В приложении также может быть реализована функциональность сохранения информации о сканированных продуктах для последующего использования или инвентаризации.

### 2. Рекомендации по внедрению в бизнес-процессы

Приложение может быть интегрировано в такие бизнес-процессы, как приемка товаров на складе, инвентаризация, маркировка активов компании и помощь в оперативном нахождении информации о товаре для персонала и клиентов в розничной торговле. Для максимальной эффективности рекомендуется связать данные, собранные с помощью приложения, с внутренней базой данных или системой управления складом.

### 3. Оценка влияния приложения на эффективность бизнес-операций

Использование моего приложения может значительно ускорить процесс работы с товарами и информацией, минимизируя ручной ввод данных и сокращая возможность человеческих ошибок. Это может привести к повышению общей производительности, оптимизации оборота запасов и улучшению качества обслуживания клиентов путем предоставления быстрой и точной информации. Чуть менее очевидное, но важное преимущество – возможность использования данных о сканировании для анализа оборота товаров и оптимизации закупок. Отмечу, что полный потенциал приложения зависит от того, насколько оно будет интегрировано в существующие IT-системы и бизнес-процессы компании.

### 4. Обеспечение безопасности и сохранности данных в возможных сценариях применения

Важным аспектом эксплуатации приложения для распознавания штрихкодов является обеспечение конфиденциальности и защиты обрабатываемой информации. Следует реализовать меры по шифрованию данных и контролю доступа, чтобы предотвратить возможные утечки и несанкционированные действия со стороны третьих лиц. Для этого необходимо настроить протоколы безопасного соединения и внедрить систему авторизации пользователей. Помимо этого, рекомендуется регулярное создание резервных копий данных, чтобы избежать их потери в случае технических сбоев или других непредвиденных ситуаций. Внедрение этих мер обеспечивает повышение доверия пользователей к приложению и гарантирование соблюдения правил конфиденциальности в настоящее время, когда информационная безопасность играет ключевую роль в любом бизнесе.

## Обзор полученных результатов

После завершения разработки и начала использования приложения было проведено углублённое тестирование и наблюдение за его функционированием в реальных условиях. Результаты показали, что приложение работает стабильно, эффективно справляется с задачами сканирования различных типов штрихкодов и демонстрирует высокую производительность.

### 1. Анализ соответствия разработанного приложения первоначальным требованиям

Приложение было разработано для предоставления лёгкого в использовании, быстрого и точного сканера штрихкодов. Исходя из отзывов пользователей и личного наблюдения за его работой, можно сделать вывод, что все первоначальные требования были выполнены. Приложение показало хорошую совместимость с разнообразными устройствами и эффективное функционирование в разных условиях освещения.

### 2. Финансовые выгоды от замены профессионального сканерного оборудования на приложение

Одним из явных преимуществ использования приложения является снижение издержек, так как нет необходимости в приобретении специализированного сканерного оборудования. Приложение можно установить на любой смартфон или планшет, что делает его доступным и экономически выгодным решением для предприятий любого размера. Это, в свою очередь, может привести к значительной экономии бюджета, особенно с учётом того, что стоимость профессионального сканирования может быть довольно велика.

### 3. Оценка удовлетворенности потребителей

На основании обратной связи от пользователей и неформальных опросов, удовлетворенность потребителей приложением кажется высокой. Пользователи ценят его за простоту использования, надёжность и эффективность. Кроме того, способность адаптироваться к различным бизнес-моделям и легкость интеграции с существующими бизнес-процессами также повышают уровень его ценности среди предпринимателей.

### 4. Гибкость настройки пользовательского интерфейса

Благодаря модульной архитектуре приложения, предоставляется возможность его кастомизации под нужды и брендинг конкретного заказчика. Пользовательский интерфейс приложения может быть адаптирован для соответствия фирменным цветам, логотипам и дизайнерским стандартам компании, улучшая тем самым пользовательский опыт и укрепляя корпоративную идентичность. Эта гибкость значительно повышает привлекательность приложения для бизнес-партнёров, так как позволяет им лучше интегрировать новый инструмент в свои работы и маркетинговые кампании, создавая положительное впечатление у их клиентов и укрепляя их лояльность.

## Выводы и предложения по дальнейшему развитию

На основе анализа данных и обратной связи от пользователей становится ясно, что мое приложение эффективно решает поставленные задачи и имеет большой потенциал для расширения функционала. Постоянное мониторинг и внесение улучшений помогут поддерживать конкурентоспособность продукта и отвечать меняющимся требованиям рынка.

Итоги практической полезности приложения

Приложение подтвердило свой потенциал практической ценности для различных бизнес-процессов. Оно может не только упростить учёт товаров и управление запасами, но и предоставить возможность быстрого доступа к информации о продуктах, что положительно скажется на уровне обслуживания клиентов.

Предложения по улучшению и добавлению новых функций

Чтобы продолжать улучшать пользовательский опыт, следующим шагом может быть разработка функций расширенной аналитики, которые помогут в анализе данных о популярности продукции и оптимизации запасов. Возможно также добавление функции синхронизации с облачными хранилищами данных и создание API для интеграции с другими системами учета и управления. Необходимо также обеспечить соответствие приложения последним требованиям безопасности, чтобы гарантировать защиту данных пользователей.

Перспективы развития проекта

В перспективе можно рассмотреть возможность использования машинного обучения для автоматизации процессов принятия решений на основе собранных данных. Также перспективным направлением может быть интеграция с устройствами дополненной реальности (AR) для создания интерактивных опытов в розничной торговле, таких как виртуальные примерочные или интерактивное моделирование магазинных полок. Расширение работы с различными видами кодировок и штрихкодов, включая новые и региональные стандарты, также поможет укрепить позиции приложения на рынке.

# Заключение

В ходе дипломного проекта была проведена разработка и детальное исследование приложения для распознавания штрихкодов. Работа, основанная на анализе теоретических материалов и последующем практическом применении полученных знаний, позволила оформить приложение с использованием фирменной стилистики образовательной платформы GeekBrains, что в свою очередь продемонстрировало понимание брендинга и его значимости в современном программном обеспечении.

Теоретические выводы подчеркнули важность удобного пользовательского интерфейса и роли брендинга в узнаваемости и доверии к приложению. Выполненное практическое исследование показало, что приложение отличается высокой точностью и скоростью распознавания штрихкодов, что делает его полезным инструментом в розничной торговле и логистике.

Оценка и анализ проведенных исследований выявили, что приложение является стабильным и функциональным решением, способным удовлетворять потребности пользователей. Дополнительно исследование показало, что методы распознавания штрихкодов, использованные в приложении, могут быть масштабированы и адаптированы для различных типов данных.

Особое внимание было уделено практической значимости проекта, подчеркивая возможности дальнейшей кастомизации приложения под специфику разных компаний. Рекомендации касались улучшения функций распознавания и внедрения дополнительных сервисов, которые могут расширять базовые возможности приложения.

Достигнутыми результатами является не только успешная разработка приложения, но также оправдание изначально поставленной гипотезы о возможности интеграции функционального решения с корпоративными элементами брендинга. Проект показал, что такие приложения не только могут стать частью брендированной среды, но и дополнить ее, создавая дополнительную ценность для компании и ее клиентов.

В целом, дипломный проект является наглядным тому примером, как теоретические знания и практические навыки могут быть успешно применены для создания функционального и в то же время привлекательного продукта. В будущем можно предположить дальнейшее углубление исследований в области улучшения алгоритмов распознавания, а также расширение функциональности с целью выхода на международный рынок и адаптации к различным бизнес-процессам.

# Список используемой литературы

* *Вебсайт: https://developer.android.com/studio/intro*
* *Вебсайт: https://ru.wikipedia.org/wiki*
* *Вебсайт: https://interid.ru/vidy-shtrih-kodov*
* *Вебсайт: https://developers.google.com/ml-kit?hl=ru*

# Приложения

## Характеристики персонального компьютера, на котором выполнялась разработка

|  |  |
| --- | --- |
| OS Name | Microsoft Windows 10 Pro |
| Version | 10.0.19045 Build 19045 |
| Other OS Description | Not Available |
| OS Manufacturer | Microsoft Corporation |
| System Name | ENIAC |
| System Manufacturer | Default string |
| System Model | Default string |
| System Type | x64-based PC |
| System SKU | Default string |
| Processor | Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz, 2601 Mhz, 10 Core(s), 20 Logical Processor(s) |
| BIOS Version/Date | American Megatrends Inc. 5.11, 11.01.2021 |
| SMBIOS Version | 3 |
| Embedded Controller Version | 255.255 |
| BIOS Mode | UEFI |
| BaseBoard Manufacturer | MACHINIST |
| BaseBoard Product | X99-K9 |
| BaseBoard Version | V2.0 |
| Platform Role | Desktop |
| Secure Boot State | Off |
| PCR7 Configuration | Binding Not Possible |
| Windows Directory | C:\WINDOWS |
| System Directory | C:\WINDOWS\system32 |
| Boot Device | \Device\HarddiskVolume1 |
| Locale | Russia |
| Hardware Abstraction Layer | Version = "10.0.19041.3636" |
| User Name | ENIAC\rmurz |
| Time Zone | Astrakhan Standard Time |
| Installed Physical Memory (RAM) | 32.0 GB |
| Total Physical Memory | 31.9 GB |
| Available Physical Memory | 8.22 GB |
| Total Virtual Memory | 53.0 GB |
| Available Virtual Memory | 7.05 GB |
| Page File Space | 21.1 GB |
| Page File | C:\pagefile.sys |
| Kernel DMA Protection | Off |
| Virtualization-based security | Running |
| Virtualization-based security Required Security Properties |  |
| Virtualization-based security Available Security Properties | Base Virtualization Support, DMA Protection |
| Virtualization-based security Services Configured |  |
| Virtualization-based security Services Running |  |
| Device Encryption Support | Reasons for failed automatic device encryption: TPM is not usable, PCR7 binding is not supported, Hardware Security Test Interface failed and device is not Modern Standby, Un-allowed DMA capable bus/device(s) detected, TPM is not usable |

## Характеристики смартфона, используемого для разработки и тестирования

|  |  |
| --- | --- |
| **Общие характеристики** | |
| Производитель | Poco |
| Модельный ряд | X6 Pro 5G |
| Цвет | серый |
| Тип | смартфон |
| Операционная система | Android |
| Количество SIM-карт | 2 |
| Тип корпуса | классический |
| Материал корпуса | пластик |
| Тип SIM-карты | Nano-SIM (12.3x8.8x0.67 мм) |
| Размеры | 74.34x160.45x8.25 мм |
| Вес | 186 гр |
| **Экран** | |
| Тип сенсорного экрана | мультитач, емкостный |
| Диагональ | 6.67 дюйм |
| Размер изображения | 2712x1220 |
| Тип экрана | AMOLED |
| Число пикселей на дюйм | 446 |
| Соотношение сторон | 20:9 |
| Частота обновления экрана | 120 Гц |
| **Мультимедийные возможности** | |
| Фронтальная камера | 16 Мп |
| Тыловая камера | 64 Мп |
| Количество мегапикселей дополнительной основной камеры | 8 + 2 Мп |
| Особенности тыловой камеры | автофокус |
| Макс. разрешение видео | 3840x2160 |
| Макс. частота кадров видео | 60 кадров/с |
| Аудио | MP3, стереодинамики |
| **Связь** | |
| Стандарт | GSM 900/1800/1900, 3G, 4G LTE, 5G |
| Поддержка диапазонов LTE | LTE 900 (B8), LTE 850 (B5), LTE 1900 (B2), LTE 2100 (B1), LTE 800 (B20), LTE 2600 (B7), LTE 1800 (B3), LTE 2600 (B38), LTE 2500 (B41), LTE 2300 (B40), LTE 700 (B17), LTE 1700 (B4), LTE 700 (B28), LTE 700 (B12), LTE 800 (B19), LTE 800 (B18), LTE 800 (B26) |
| Интерфейсы | IRDA, NFC, 802.11ax, Bluetooth 5.4 |
| Спутниковая навигация | BeiDou, A-GPS, Galileo, ГЛОНАСС, GPS, QZSS |
| **Память и процессор** | |
| Процессор | MediaTek Dimensity 8300 Ultra |
| Частота процессора | 3350 МГц |
| Количество ядер процессора | 8 |
| Видеопроцессор | Mali-G615 MC6 |
| Оперативная память | 12 Гб |
| Встроенная память | 512 Гб |
| **Питание** | |
| Тип аккумулятора | Li-polymer |
| Аккумулятор | несъемный |
| Емкость аккумулятора | 5000 мАч |
| Тип разъема для зарядки | USB Type-C |
| Функция быстрой зарядки | есть |

## Ключевые возможности библиотеки Google ML Kit

Читает большинство стандартных форматов.

Линейные форматы: Codabar, Code 39, Code 93, Code 128, EAN-8, EAN-13, ITF, UPC-A, UPC-E.

2D-форматы: Aztec, Data Matrix, PDF417, QR-код.

Автоматическое определение формата

Сканируйте все поддерживаемые форматы штрих-кодов одновременно без необходимости указывать формат, который вы ищете, или увеличьте скорость сканирования, ограничив детектор только теми форматами, которые вас интересуют.

Извлекает структурированные данные

Структурированные данные, хранящиеся в одном из поддерживаемых 2D-форматов, анализируются автоматически. Поддерживаемые типы информации включают URL-адреса, контактную информацию, события календаря, адреса электронной почты, номера телефонов, приглашения к SMS-сообщениям, номера ISBN, информацию о подключении Wi-Fi, географическое местоположение и информацию о драйверах в соответствии со стандартом AAMVA.

Работает с любой ориентацией

Штрих-коды распознаются и сканируются независимо от их ориентации: правой стороной вверх, вверх ногами или вбок.

Запускается на устройстве

Сканирование штрих-кода выполняется полностью на устройстве и не требует подключения к сети.

Обратите внимание, что этот API не распознает штрих-коды в следующих формах:

1D штрих-коды, состоящие только из одного символа

Штрих-коды в формате ITF с количеством символов менее шести, известно, что этот формат ненадежен из-за отсутствия контрольной суммы.

Штрих-коды, закодированные с помощью FNC2, FNC3 или FNC4.

QR-коды, созданные в режиме ECI

Этот API распознает не более 10 штрих-кодов за один вызов API.